

# **Transmiterea datelor de la senzori prin intermediul LoraWan**

Tomescu Ilie Alexandru

## Cuprins

1. Introducere .....	3
2. Tehnologie utilizată .....	4
2.1. LoRa / LoRaWAN .....	4
2.2. The Things Network .....	4
2.3. Thingspeak .....	5
2.4. Microsoft Azure .....	6
2.5. Node-RED [5] .....	6
3. Metodologie .....	7
3.1. Hardware folosit .....	7
3.2. Conexiuni .....	9
3.3. The Things Network setup .....	10
3.4. Node-RED API .....	11
3.5. Date colectate Thingspeak .....	13
4. Concluzii și planuri de viitor .....	15
5. Referințe .....	17

## 1. Introducere

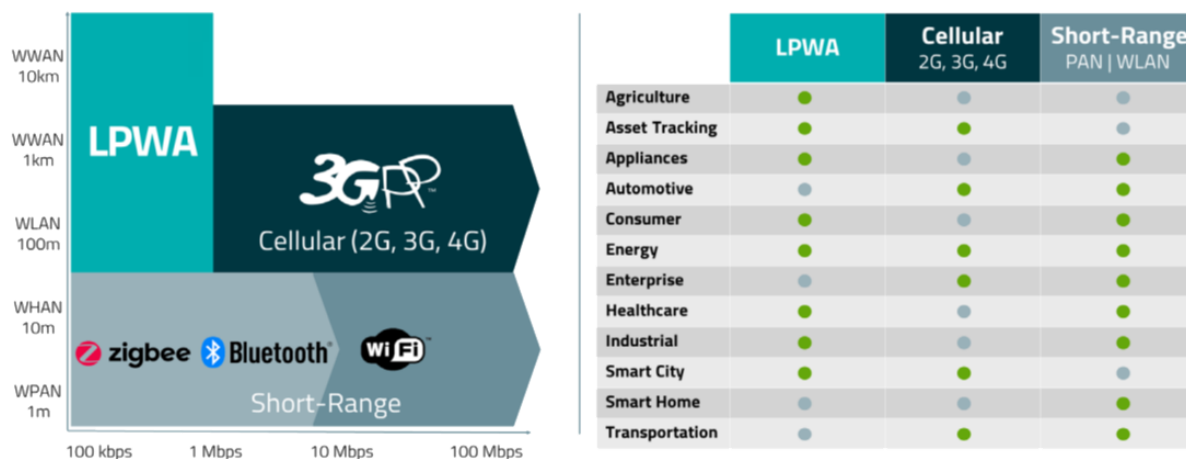
Internet of Things (IoT) reprezintă un ecosistem de dispozitive conectate care pot schimba date printr-o conexiune cablată sau wireless. Aceste dispozitive pot fi telefoane inteligente, laptopuri, electronice inteligente sau dispozitive care au atașați senzori, categorii în care se situează și proiectul de față. Aceste ecosisteme IoT pot fi atât unele pe arii restrânse, cum ar fi case sau birouri, sau pe suprafețe mai mari, cum ar fi orașele.

Prima rețea de acest fel a fost ARPANET, o rețea universitară creată de câțiva studenți și un cercetător de la universitatea Carnegie Mellon pentru a monitoriza starea unui distribuitor de suc [1].

De atunci și până în prezent industria IoT a devenit parte din viața de zi cu zi a noastră deoarece suntem obișnuiți ca să avem în jurul nostru dispozitive pe care le putem accesa și de la care să primim date aproape instant.

Un domeniu care a prezentat rezultate excelente prin adoptarea dispozitivelor IoT este Agricultură, unde cu ajutorul unor sonde care monitorizează anumite lucruri precum umiditatea solului, compoziția chimică a pământului sau condițiile de iluminare conectate prin interfețe IoT pot trimite date care să fie accesate de oriunde sau pe baza acestora se poate automatiza pornire și oprirea unor sisteme de irigații.

Însă în domenii ca acesta nu se poate asigura întotdeauna accesul la o rețea electrică, astfel a apărut necesitatea unor standarde de comunicație wireless pe distanțe mari, cu consum mic de curent. O astfel de tehnologie este LPWA (low-power, wide-area) care sacrifică cantitatea de date care poate fi transmisă pentru a menține un consum redus și a acoperii distanțe mari (kilometri sau zeci de kilometri).



Scopul general al acestei lucrări este de a folosi LoRa, care este o tehnologie de tipul LPWA, pentru a transmite date provenite de la senzori pe diferite distanțe prin intermediul unui gateway conectat la The Things Network și mai apoi stocarea lor într-o platformă IoT specializată precum Thingspeak.

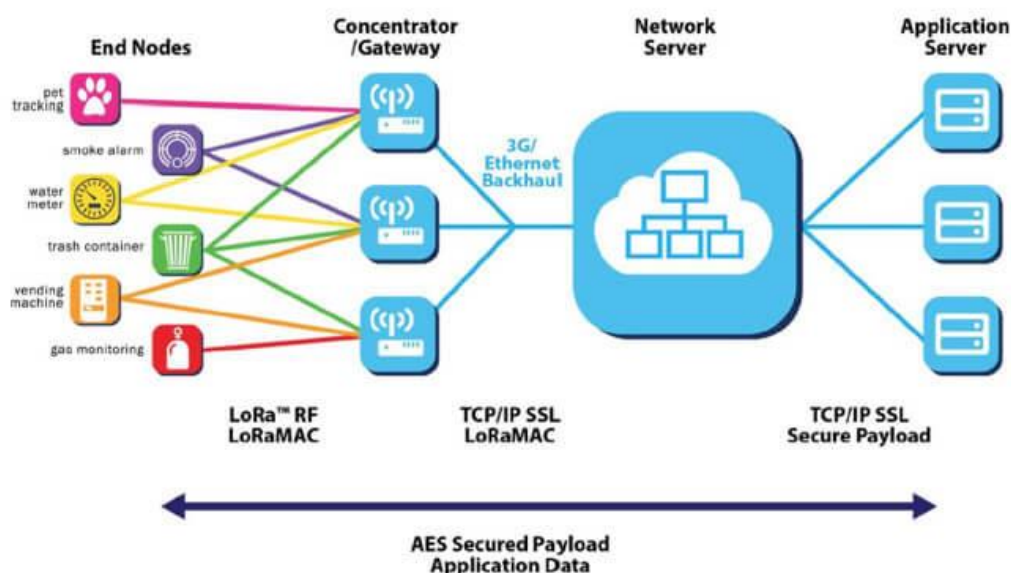
## 2. Tehnologie utilizată

### 2.1. LoRa / LoRaWAN

Cum am menționat anterior, LoRa este o categorie de comunicații radio caracterizate prin distanță mare de transmisie cu un consum mic de curent. Dispozitivele LoRa dezvoltate de Semtech și tehnologia de transmisie radio wireless au devenit o tehnologie foarte folosită deoarece au o aplicabilitate extraordinară în rețele de senzori wireless, IoT și crearea de rețele de dispozitive inteligente [2].

Specificația deschisă LoRaWAN este un protocol de rețea low-power, wide-area (LPWAN) bazat pe tehnologia LoRa. Conceput pentru a conecta fără fir la Internet of Things cu baterii în rețele regionale, naționale sau globale, protocolul LoRaWAN valorifică spectrul radio fără licență din banda industrială, științifică și medicală (ISM) [2].

#### Architecture



### 2.2. The Things Network

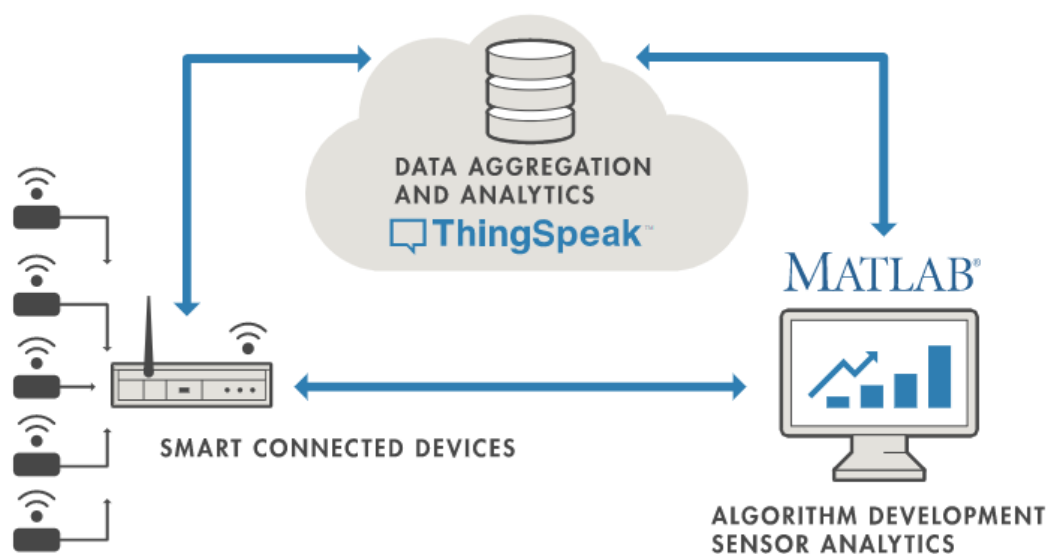
The Things Network (TTN) este o inițiativă lansată de societatea civilă olandeză. Acesta își propune să fie instalate rețele LoRaWAN în fiecare oraș din întreaga lume, iar prin interconectarea acestor rețele locale, TTN vrea să construiască o infrastructură la nivel mondial pentru a facilita un Internet of Things public [3].

At this moment, there are 18990 gateways up and running



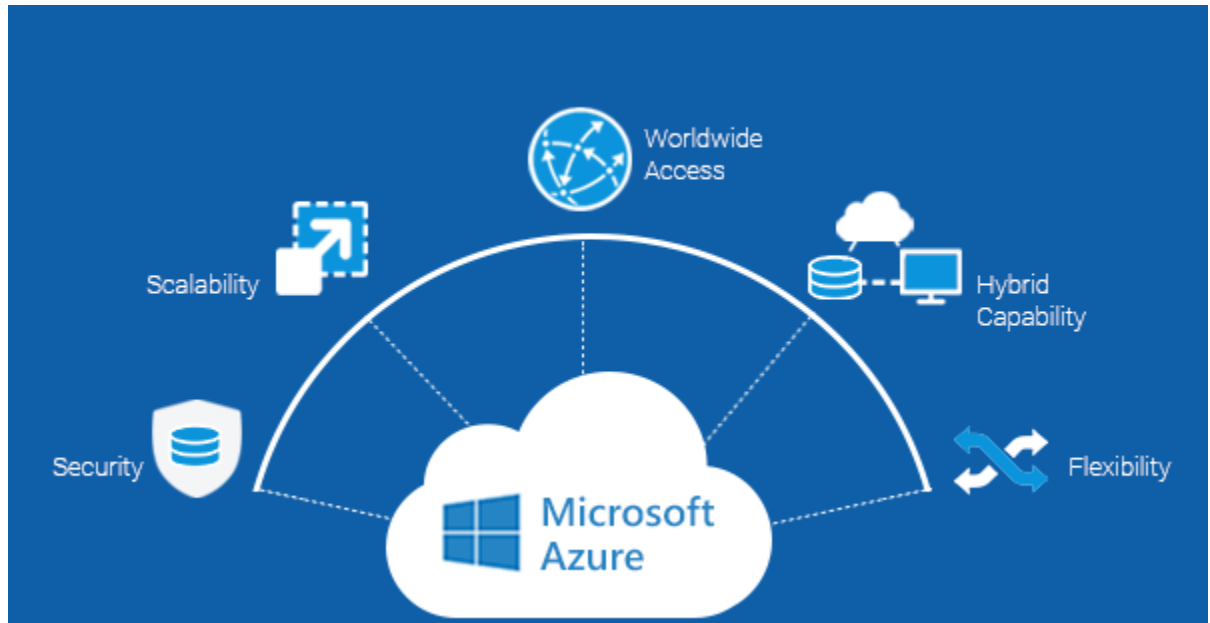
### 2.3. Thingspeak

Este o platformă IoT care permite vizualizarea și analiza unor fluxuri de date în cloud. ThingSpeak oferă vizualizări instantanee ale datelor postate de dispozitivele, cu posibilitatea de a executa cod MATLAB pentru a prelucra aceste date pe măsură ce vin. Această platformă este adesea utilizată pentru prototipare și dovada conceptului sistemelor IoT care necesită analize [4].



## 2.4. Microsoft Azure

Microsoft Azure este un serviciu de cloud computing creat de Microsoft pentru construirea, testarea, implementarea și gestionarea aplicațiilor și serviciilor prin intermediul centrelor de date gestionate de Microsoft. Oferă software ca serviciu (SaaS), platformă ca serviciu (PaaS) și infrastructură ca serviciu (IaaS) și acceptă mai multe limbaje, instrumente și cadre de programare diferite, inclusiv software și sisteme specifice Microsoft și terțe părți.



## 2.5. Node-RED [5]

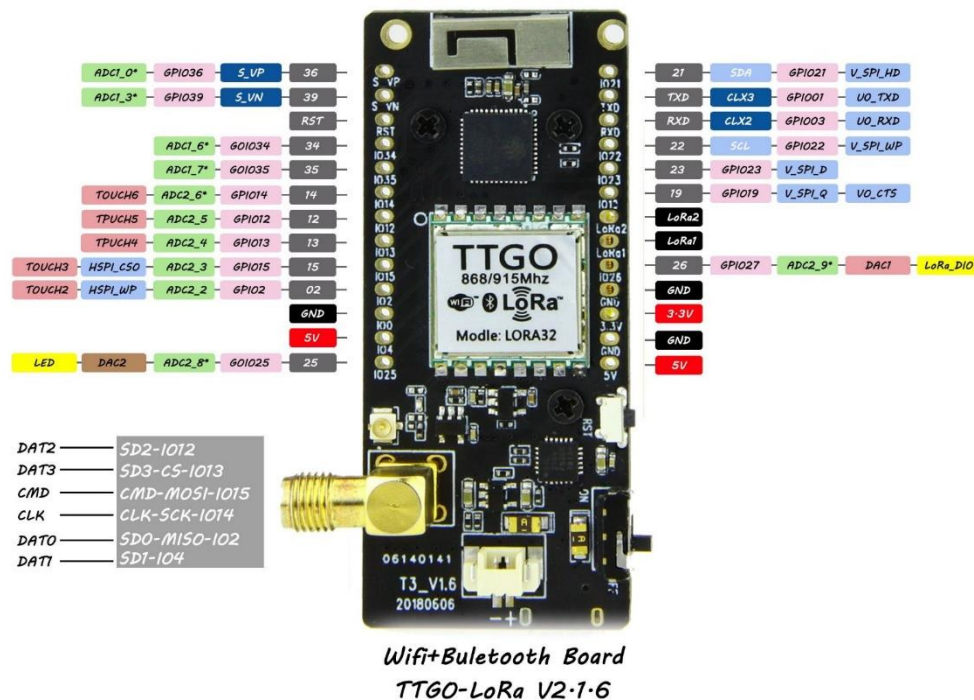
Node-RED este un instrument de programare pentru conectarea între dispozitive hardware, API-uri și servicii online în moduri noi și interesante.

Oferă un editor bazat pe browser care facilitează conectarea fluxurilor împreună utilizând gama largă de noduri din paletă care pot fi implementate în timpul rulării sale cu un singur clic. Este construit pe Node.js, profitând din plin de modelul său blocat de evenimente. Acest lucru îl face ideal pentru a rula la marginea rețelei pe hardware ieftin sau în cloud.

### 3. Metodologie

#### 3.1. Hardware folosit

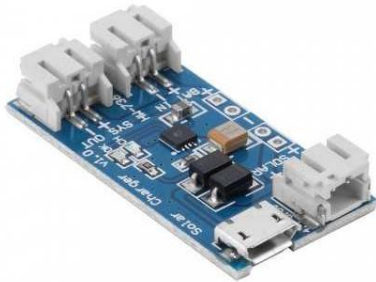
##### 1. TTGO LoRa32\_V2.1-1.6



Am ales această plăcuță de dezvoltare deoarece are integrate o selecție de componente ce facilitează dezvoltarea în mediu LoRa:

- ESP32 – este un microcontroler low-cost, care oferă conectivitate Bluetooth și Wi-Fi. O altă caracteristică importantă este consumul redus atât în modul de operare normală cât și în modurile de sleep.
- Modul radio LoRa – folosit pentru comunicația LoRa
- Ecran Oled SSD1306 – care poate fi folosit atât pentru a afișa informații la Runtime sau mesaje pentru debug
- Antena Radio
- Conector și circuit de încărcare baterie LiPo

2. Modul de încărcare solara  
acumulator 3.7 V, CN3065



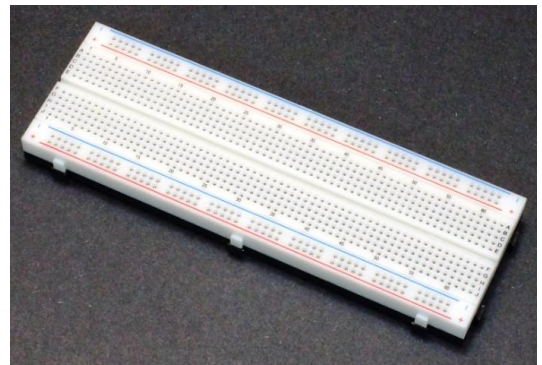
6. Baterie LiPo 3.7V 1800mh



3. Panou solar 5V 1W



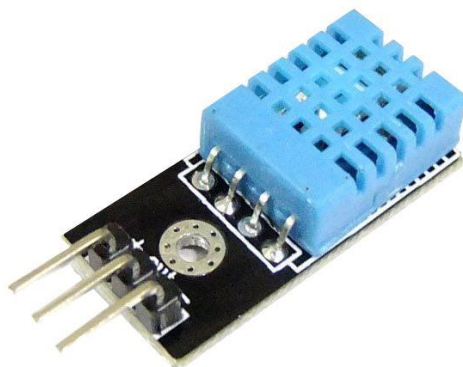
7. Breadboard



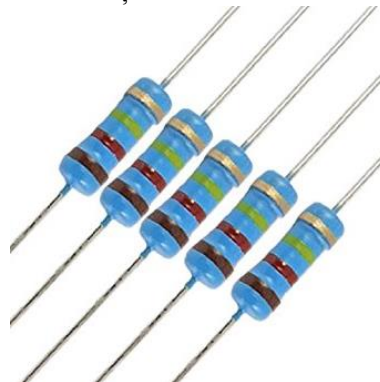
4. Senzor hidrometric capacitiv



5. Senzor de temperatură și umiditate,  
DHT 11



8. Rezistențe de diferite valori

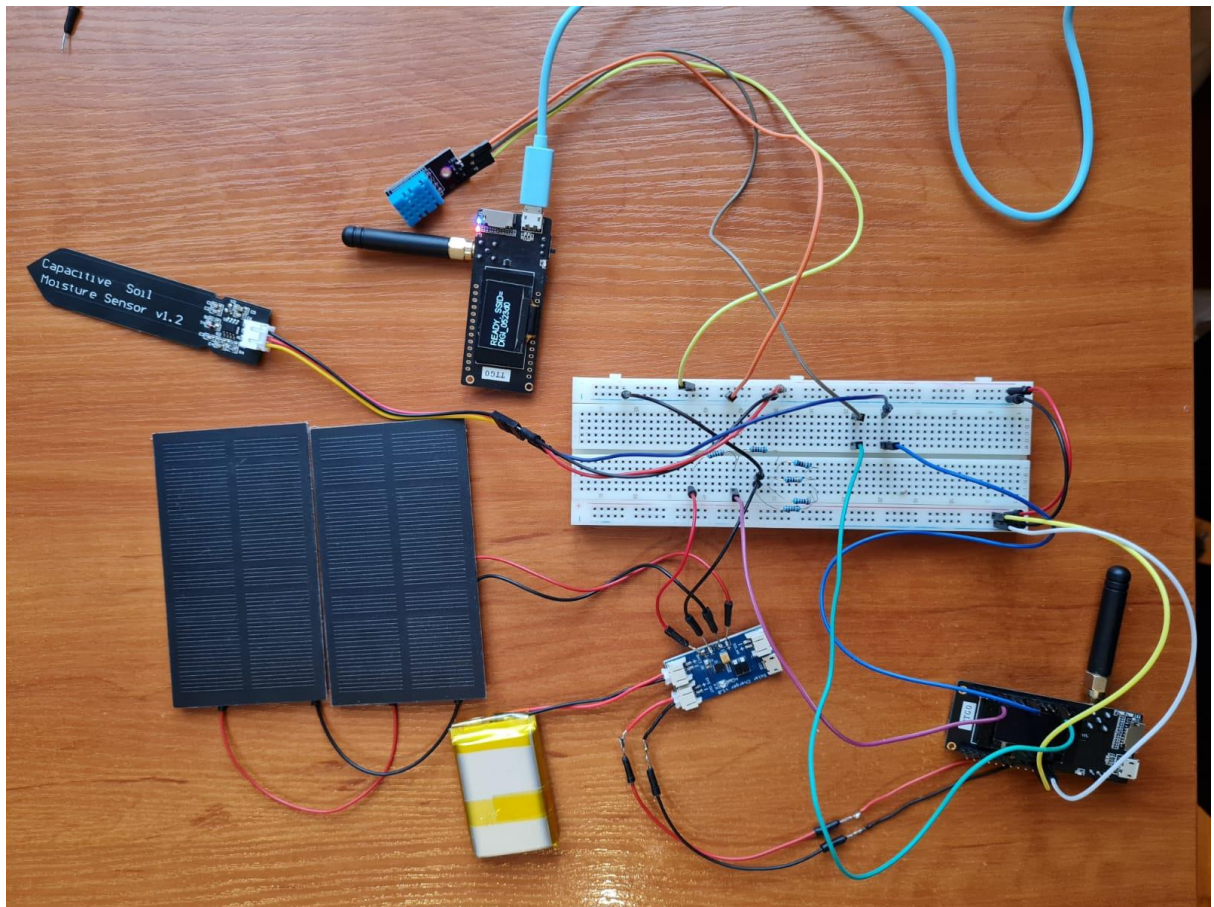


9. Fire Tată-Tată/Mamă-Tată





### 3.2. Conexiuni



O plăcuță TTGO LoRa32 acționează ca și gateway pentru The Things Network. Pentru aceasta am folosit scriptul de la [6] prin care am putut să o fac să se comporte ca un gateway care ascultă pe un singur canal. Problema cu această implementare este faptul că LoRa folosește 8 canale, astfel pachetele care nu sunt trimise pe primul canal se pierd.

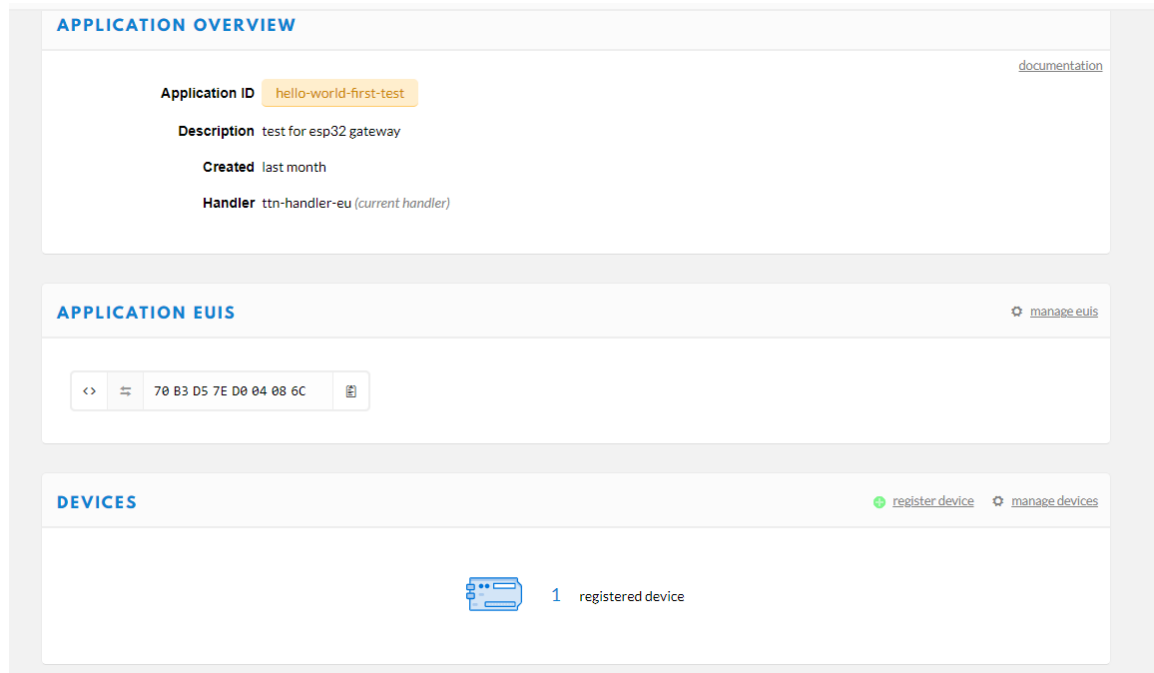
O a doua plăcuță TTGO LoRa32 care are rolul de a trimite datele primite de la senzori către TTN prin gateway. Între plăcuță și celelalte componente am făcut următoarele conexiuni:

- 3.3 V și GND de la plăcuță sunt legate la +, respectiv – de la breadboard, loc în care sunt legate și celelalte terminale de VCC și GND de la senzori.
- D4 – Data pin DHT 11
- D13 – Data pin Senzor hidrometric capacitiv
- SYS OUT CN3065 – Conector LiPo TTGO LoRa32
- BATT IN CN3065 - Conector baterie LiPo 3.7 V
- SOLAR CN3065 – 2 Panouri solare 5V 1W în paralel
- BATT IN CN3065 – Divizor Tensiune  $R1 = 2k \Omega$ ,  $R2 = 7.33k \Omega$  – D34 TTGO LoRa32

Toate datele citite sunt citite de plăcuță și trimise către gateway, după care aceasta v-a intra în deep sleep timp de 5 minute pentru a reduce consumul de curent între transmisii. Modul în care sunt citite datele și transmise este prezentat în [7].

### 3.3. The Things Network setup

În consola din TTN am creat o aplicație la care am adăugat un device, acesta fiind chiar plăcuța TTGO LoRa32 care transmite date.



**APPLICATION OVERVIEW**

Application ID: `hello-world-first-test` [documentation](#)

Description: test for esp32 gateway

Created: last month

Handler: ttn-handler-eu (current handler)

**APPLICATION EUIs** [manage euis](#)

`<> 70 B3 D5 7E D0 04 08 6C`

**DEVICES** [register device](#) [manage devices](#)

1 registered device

Datele primite trebuie decodate pentru că acestea sunt trimise ca și numere în baza 16, iar pentru acest lucru vom folosi următoarea funcție de decode.



decoder converter validator encoder [remove decoder](#)

```
1 function Decoder(bytes, port) {
2   // Decode plain text; for testing only
3   var temperatura = bytes[2]<<24>>16 | bytes[3];
4   var umiditate = bytes[0]<<8 | bytes[1];
5   var voltaj = bytes[4]<<8 | bytes[5];
6   var umiditate_sol = bytes[6]<<8 | bytes[7];
7
8   temperatura = temperatura / 100;
9   umiditate = umiditate / 100;
10  voltaj = voltaj/100;
11  umiditate_sol = umiditate_sol/100;
12
13  return {
14    field1: temperatura,
15    field2: umiditate,
16    field3: voltaj,
17    field4: umiditate_sol,
18  }
19 }
```

decoder has no changes

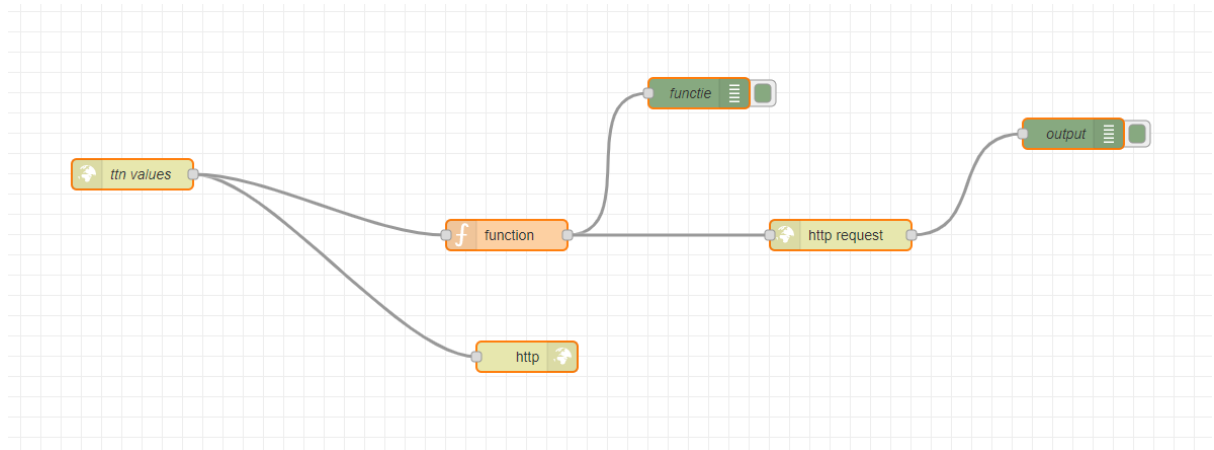
Deoarece TTN-ul nu stochează datele vom folosi HTTP Integration pentru a trimite toate datele mai departe către API-ul creat cu Node-RED. Datele sunt trimise în următorul format JSON

```
{
  "app_id": "my-app-id",           // Same as in the topic
  "dev_id": "my-dev-id",          // Same as in the topic
  "hardware_serial": "0102030405060708", // In case of LoRaWAN: the DevEUI
  "port": 1,                      // LoRaWAN FPort
  "counter": 2,                   // LoRaWAN frame counter
  "is_retry": false,              // Is set to true if this message is a retry (you could a
  "confirmed": false,             // Is set to true if this message was a confirmed message
  "payload_raw": "AQIDBA==",      // Base64 encoded payload: [0x01, 0x02, 0x03, 0x04]
  "payload_fields": {},           // Object containing the results from the payload functio
  "metadata": {
    "time": "1970-01-01T00:00:00Z", // Time when the server received the message
    "frequency": 868.1,             // Frequency at which the message was sent
    "modulation": "LORA",           // Modulation that was used - LORA or FSK
    "data_rate": "SF7BW125",       // Data rate that was used - if LORA modulation
    "bit_rate": 50000,              // Bit rate that was used - if FSK modulation
    "coding_rate": "4/5",          // Coding rate that was used
    "gateways": [
      {
        "gtw_id": "ttn-herengracht-ams", // EUI of the gateway
        "timestamp": 12345,              // Timestamp when the gateway received the message
        "time": "1970-01-01T00:00:00Z", // Time when the gateway received the message - left o
        "channel": 0,                   // Channel where the gateway received the message
        "rssi": -25,                   // Signal strength of the received message
        "snr": 5,                      // Signal to noise ratio of the received message
        "rf_chain": 0,                 // RF chain where the gateway received the message
        "latitude": 52.1234,           // Latitude of the gateway reported in its status upda
        "longitude": 6.1234,          // Longitude of the gateway
        "altitude": 6                  // Altitude of the gateway
      },
      //...more if received by more gateways...
    ],
    "latitude": 52.2345,              // Latitude of the device
    "longitude": 6.2345,              // Longitude of the device
    "altitude": 2                     // Altitude of the device
  },
  "downlink_url": "https://integrations.thethingsnetwork.org/ttn-eu/api/v2/down/my-app-id/my-pr
}
```

### 3.4. Node-RED API

Pentru a păstra doar datele de interes din JSON-ul prezentat anterior am folosit Node-RED.

Acesta rulează pe un Server Ubuntu care se află într-o mașină virtuală creată prin intermediul Microsoft Azure. Pentru ca acesta să poată primi rula Node-Red a fost nevoie de adăugarea unei Inbound Security Rule care să permită accesul prin portul 1880 prin TCP.



Blocurile din aceasta diagrama au urmatoarele funcții:

1. Primul bloc, *ttn values*, așteaptă un request HTTP POST din care preia payload-ul și îl trimite mai departe.
2. Blocul *http* trimite un răspuns înapoi catre sursa de la care a venit requestul.
3. Blocul *function* este o funcție în JavaScript care preia payload-ul, parsează datele pe care dorim să le trimitem către Thingspeak și creează URL-ul pentru un request HTTP GET

```

1
2 var rssi = msg.payload.metadata gateways[0].rssi
3 var field1 = msg.payload.payload_fields.field1;
4 var field2 = msg.payload.payload_fields.field2;
5 var field3 = msg.payload.payload_fields.field3;
6 var field4 = msg.payload.payload_fields.field4;
7 msg.url = "https://api.thingspeak.com/update.json?api_key=BPR43R1KY5RSJIG5"
8 + "&field1=" + field1
9 + "&field2=" + field2
10 + "&field3=" + field3
11 + "&field4=" + field4
12 + "&field5=" + rssi;
13 msg.payload=msg.url;
14 return msg;

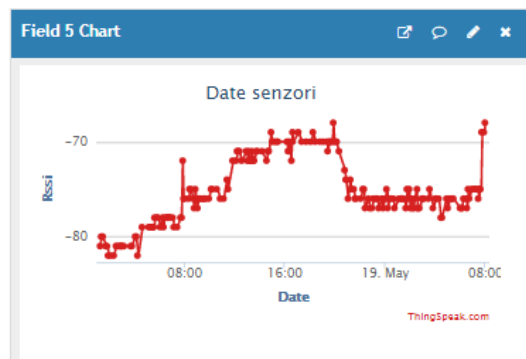
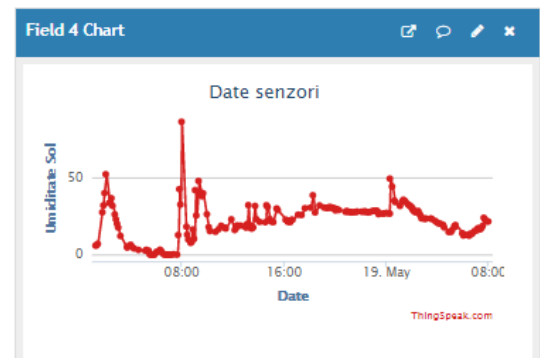
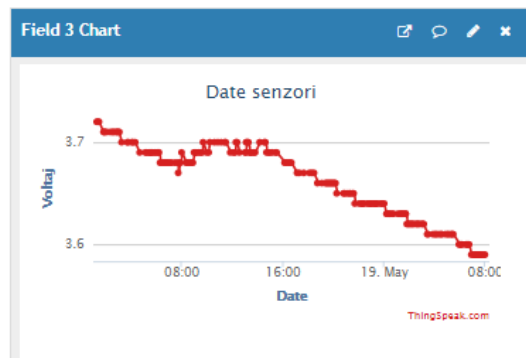
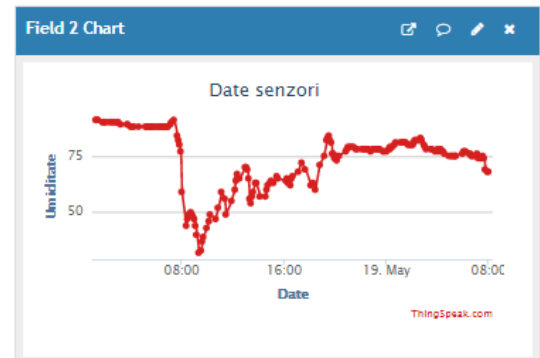
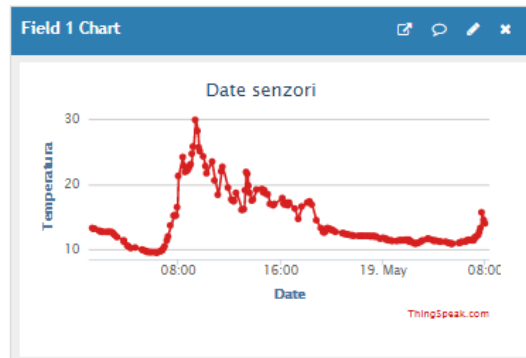
```

4. Blocul *http request* preia URL-ul și efectuează un request HTTP GET
5. Cele 2 blocuri verzi sunt blocuri de debug care afișează în consola de debug output-urile de la celelalte blocuri.

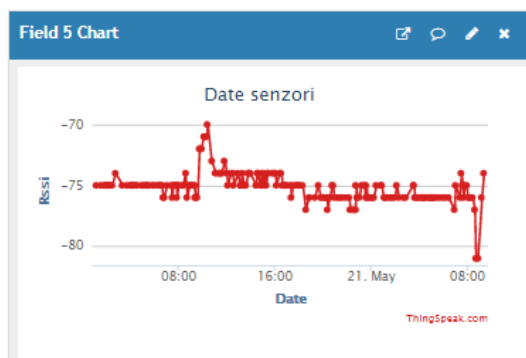
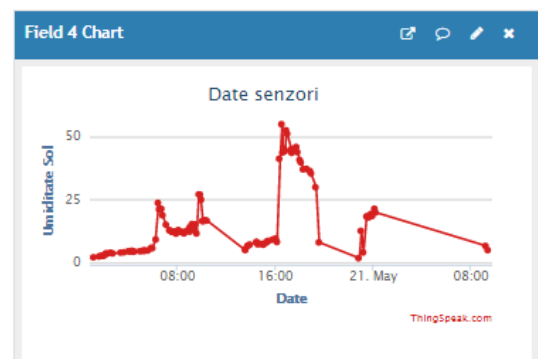
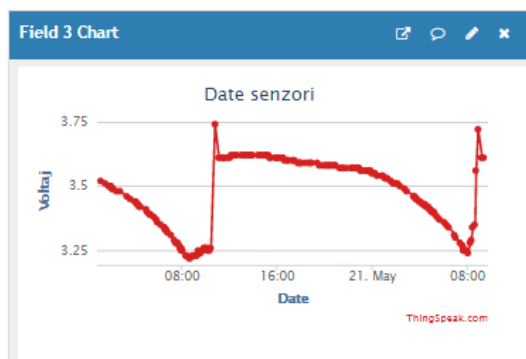
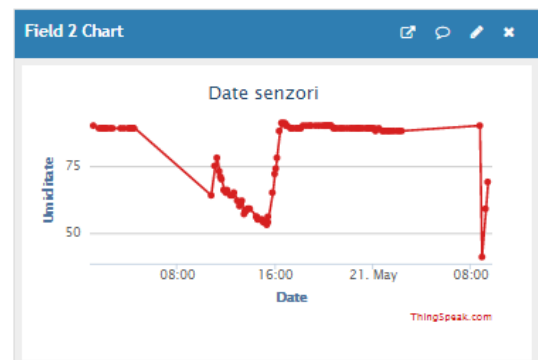
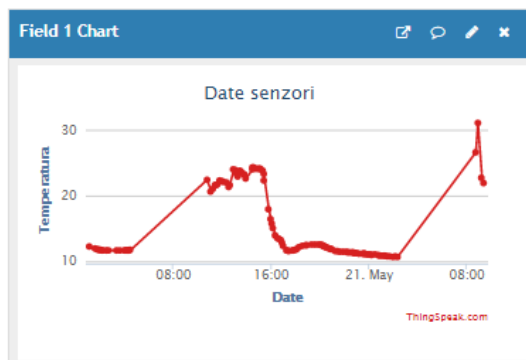
### 3.5. Date colectate Thingspeak

În cadrul acestui proiect am făcut teste pe mai multe distanțe pentru a testa atât puterea semnalului cât și funcționare corectă a senzorilor.

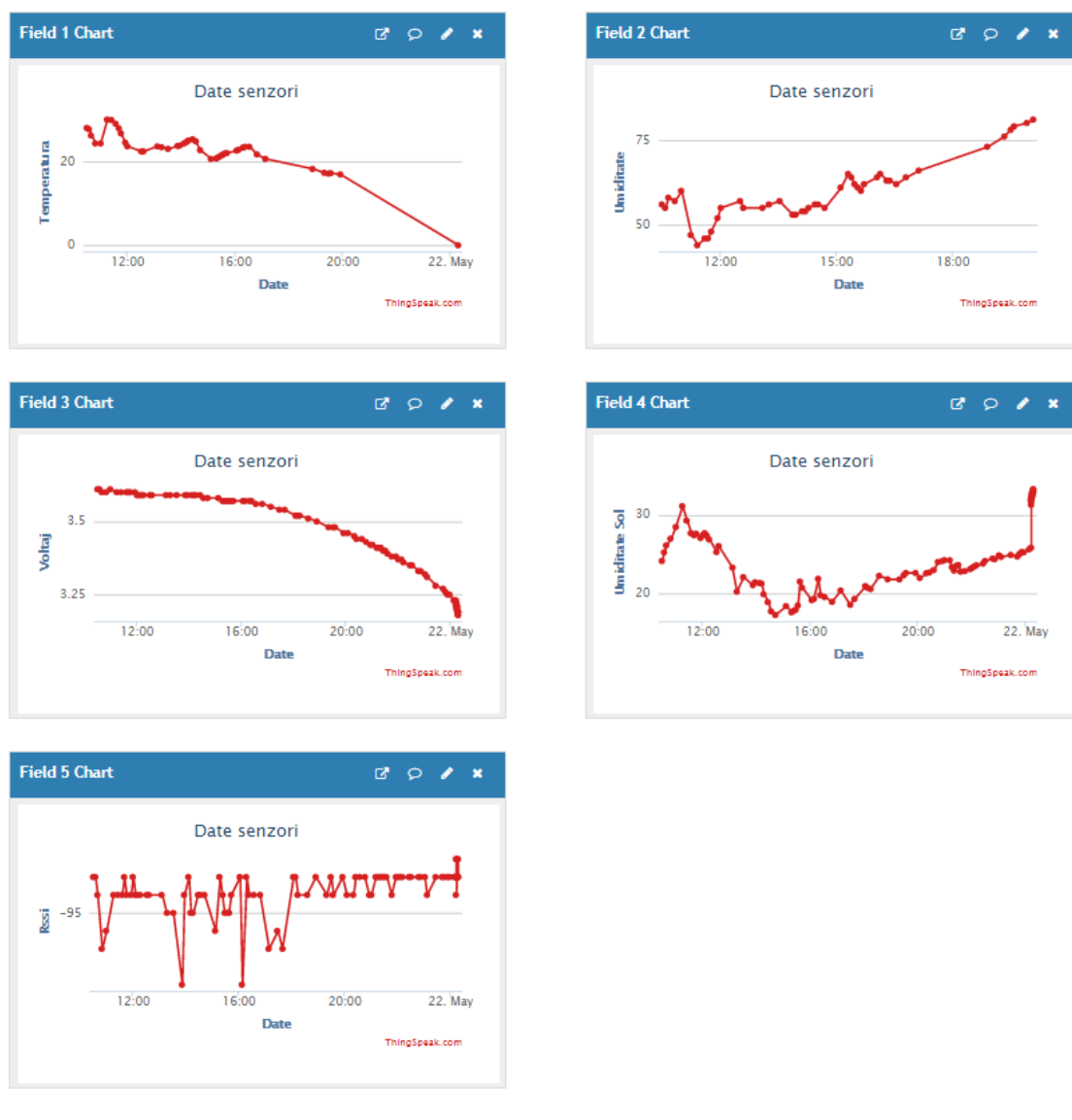
#### 1. Test 15 metri



## 2. Test 40 metri



### 3. Test 70 metri



### 4. Concluzii și planuri de viitor

Scopul acestui proiect a fost de a învăța cum funcționează comunicația LoRa, cum se pot prelua date în TTN și cum se pot trimite aceste date pentru a fi prelucrate și stocate. Consider că aceste puncte au fost îndeplinite și astfel am creat o bază pentru a extinde acest proiect la unul mai complex. Link-ul către un scurt demo este [8]

Ca și planuri de viitor ar fi următoarele:

- Tranziția către un gateway cu opt canale pentru a evita pierderea pachetelor și astfel se pot adăuga mai multe noduri care să trimită date.
- Folosirea unei plăcuțe de dezvoltare care să consume mai puțin curent pentru a maximiza durata de funcționare. O astfel de plăcuță este cea oferită de Heltec, care e bazată pe ASR605x (ASR6501, ASR6502) [9].

- Folosirea unui circuit integrat pentru monitorizarea voltajului bateriei care oferă o acuratețe mai bună și poate duce și la minimizarea circuitului,
- Experimentarea cu mai mulți senzori pentru colectarea datelor
- Efectuarea de teste pe distanțe mai mari și astfel testarea limitelor acestei transmisii
- Sudarea conexiunilor pentru a nu apărea probleme de conectivitate



## 5. Referințe

- [1] “Evolution of Internet of Things (IoT): Past, present and future.” <https://www.techaheadcorp.com/knowledge-center/evolution-of-iot/> (accessed May 16, 2021).
- [2] “What is LoRa? | Semtech LoRa Technology | Semtech.” <https://www.semtech.com/lora/what-is-lora> (accessed May 16, 2021).
- [3] A. Offerman, “The Things Network sets up a world-wide Internet of Things (The Things Network) | Joinup,” Jun. 01, 2016. <https://joinup.ec.europa.eu/collection/egovernment/document/things-network-sets-world-wide-internet-things-things-network> (accessed May 16, 2021).
- [4] “Learn More - ThingSpeak IoT.” [https://thingspeak.com/pages/learn\\_more](https://thingspeak.com/pages/learn_more) (accessed May 16, 2021).
- [5] “Node-RED.” <https://nodered.org/> (accessed May 16, 2021).
- [6] “things4u/ESP-1ch-Gateway: Version 6 of the single channel gateway.” <https://github.com/things4u/ESP-1ch-Gateway> (accessed May 16, 2021).
- [7] “Tomescu-Alexandru/Transmisie\_Date\_LoRa.” [https://github.com/Tomescu-Alexandru/Transmisie\\_Date\\_LoRa](https://github.com/Tomescu-Alexandru/Transmisie_Date_LoRa) (accessed May 16, 2021).
- [8] “Demo Proiect,” [Online]. Available: <https://drive.google.com/file/d/1IVq2a-mBM8DQwcOvimy-0PZdhDAy3xB1/view?usp=sharing>.
- [9] “CubeCell – Dev-Board – Heltec Automation.” <https://heltec.org/project/htcc-ab01/> (accessed May 19, 2021).